09017232 刘晓臻_课程研学报告（一）

学号：09017232　　姓名：刘晓臻　　日期：2020.04.18

一、建立文档库

按照要求，至少要有 30 篇以上的文档。在本人的《关于雅虎搜索(Yahoo! Search)搜索引擎的学习报告》中第三部分，我所选择的论文为《Effective Online Knowledge Graph Fusion》，其参考文献只有 20 篇，故额外在其参考文献中最新的文章《Biperpedia: An ontology for search applications》的参考文献列表中选择了前 15 篇文献添加至文档库。

文档库参见文件“docs.txt”，每行为一个文档。

二、建立索引

对文档作预处理（包括排除停用词与字母统一为小写）后，使用哈希表存储 term，得到倒排索引，其中每个 posting 中还包含了该 term 在文档中出现的位置（第几个 token）。在倒排索引实现时，为了能够构建向量查询中的 term-document 矩阵，对每个 term 编号（顺序为在字典中的存储顺序）。为了显示结果，倒排索引打印到了“inverted_index.txt”中。如图 1，输出格式为：

`term: term 编号 {文档号: [该文档中第 1 次出现位置，第 2 次出现位置...], ...}`

```
2006:   51  {11: [48]}
acquisition:    119 {33: [10]}
alignment:  48  {11: [19], 16: [42], 18: [14]}
analysis:   26  {5: [71]}
applications:   29  {6: [35]}
approach:   41  {9: [48]}
architecture:   89  {24: [10]}
automatic:  118 {33: [0]}
aware:  13  {3: [12]}
based:  24  {5: [47], 8: [46], 10: [26]}
biperpedia: 27  {6: [0]}
bridges:    3   {0: [24]}
building:   2   {0: [15]}
changes:    18  {4: [31]}
collaboratively:    77  {22: [12]}
computing:  20  {5: [0]}
conquer:    40  {9: [40]}
consolidation:  32  {7: [41]}
coreference:    12  {3: [0], 32: [7]}
corpora:    122 {33: [50]}
created:    78  {22: [28]}
cross:  59  {16: [0]}
data:   34  {8: [18, 63], 12: [7], 15: [38], 17: [32], 20: [30], 21: [37], 25: [27], 27: [14]}
database:   80  {22: [42], 30: [31]}
dbpedia:    73  {21: [0]}
divide: 39  {9: [29]}
easy:   94  {25: [11]}
effective:  123 {34: [3]}
efficient:  95  {25: [17]}
electronic: 107 {30: [12]}
```

图 1

由于使用哈希表时借用了 Python 中内置的字典数据结构（其内部实现就是使用的哈希表，且是开散列），其实现查询 term 的算法的伪代码如下：

```
# string 是待查找的term, dicti 为待查找词典(hash 表)
def lookup(string, dicti):
    # hash(...)为选用的的哈希函数，将字符串映射到唯一的一个数。
    h = hash(string)
    # dicti.slots[]为该hash 表的hash 槽，若其非空则在其中存放了
    # 实际存放的term 的hash 值，及对应的postings 的指针(value)
    if (dicti.slots[h].is_empty()):
        throw(key_not_found_error)
    # 若匹配上，则找到，返回对应的值。
    elif (dicti.slots[h].actual_hash_value == h):
        return dicti.slots[h].value
    # 否则，顺序往后查找，找到底则回到最初，直到再次回到该hash 值，说明没有找到，报错
    else:
        current_slot_number = h + 1
        while(True):
            if (dicti.slots[h].is_empty()):
                throw(key_not_found_error)
            elif (dicts.slots[current_slot_number].actual_hash_value == h):
                return dicti.slots[current_slot_number].value
            else:
                if (current_slot_number + 1 >= dicti.capacity):
                    current_slot_number = 0
                elif (current_slot_number + 1 == h):
                    throw(key_not_found_error)
                else:
                    current_slot_number += 1
```

三、建立文档向量

文档是按照“doc.txt”中的顺序编号的，term 编号见“inverted_index.txt”。得到的文档向量输出在了“document_term_matrix.txt”，如图 2。每一行为一个文档向量，每个 term 的权重为其在文档中的出现次数。

图 2

四、建立查询和查询向量

查询为“Effective Online Knowledge Graph Fusion”，查询向量输出在“query_vector.txt”中。同样，每个 term 的权重为其在文档中出现的次数。需要注意的是，query 中存在 term 词典中没有的词，会被自动过滤掉。

五、利用向量模型，计算查询 q 与所有文档的相关度

为了加速计算，不直接使用向量相乘，且仅对每个 query 中的 term 逐一计算各项分数（因为不在 query 中的 term 不可能影响到排序，故可以省略对其的计算。即，使用类似于图 3 中的算法（除了相乘的不是 wf 之外），通过计算图 4 中的各项值，得到每个文档的相关度。

```
CosineScore(q)
    float Scores[N]={0}
    Initialize Length[N]
    for each query term t
    do calculate w_{t,q} and postings list for t
        for each pair(d,tf_{t,d}) in postings list
        do Scores[d] += wf_{t,d} × w_{t,q}
    Read Length[d]
    for each d
    do Scores[d] = Scores[d] / Length[d]
    return Top K of Scores[]
```

图 3

	query					document			
word	tf	wf	df	idf	q_i = wf-idf	tf	wf	d_i = normalized wf	$q_i \cdot d_i$
digital			10,000						
video			100,000						
cameras			50,000						

图 4

对于 query 的 term 权重的计算，采用教材（英文版）第六章 P128. 的增广权重：

$$\mathrm{wf}_{t,q} = 0.5 + \frac{0.5 \times \mathrm{tf}_{t,q}}{max_t(\mathrm{tf}_{t,q})}$$

对于 document 的 term 权重计算，使用 PPT 中的计算方法：

$$\mathrm{wf}_{t,d} = \frac{\mathrm{tf}_{t,d}}{max_t(\mathrm{tf}_{t,d})} \times \mathrm{idf}_t$$

另外，对于 query，采用 wf-idf 作为最终的 q_i，对于文档，作余弦归一化（对 wf 向量）。

query 一栏的计算和 document 一栏的计算过程分别输出为“query_calculation.txt”和“docs_calculation.txt”，文档中每行的数字顺序分别和图 4 中的顺序对应。

最后得到所有文档和 query 的相似度得分，排名后输出到“vector_space_rank_and_scores.txt”中，如下图 5：

```
Online data fusion  14.595991968710033
Freebase: a collaboratively created graph database for structuring human knowledge  8.938183153275219
An effective semantic search technique using ontology   5.160462449609626
Cross-lingual entity matching and infobox alignment in wikipedia    0.0
Learning to link with wikipedia 0.0
Topic indexing with wikipedia   0.0
Result of ontology alignment with RiMOM at OAEI 2006    0.0
Logmap 2.0: towards logic-based, scalable and interactive ontology matching 0.0
Matching large ontologies: A divide-and-conquer approach    0.0
Heterogeneous web data search using relevance-based on the fly data integration 0.0
Federated entity search using on-the-fly consolidation  0.0
Biperpedia: An ontology for search applications 0.0
Computing semantic relatedness using wikipedia-based explicit semantic analysis 0.0
Inconsistencies, negations and changes in ontologies    0.0
Coreference aware web object retrieval  0.0
Repeatable and reliable semantic search evaluation  0.0
Entity recommendations in web search    0.0
Ad-hoc object retrieval in the web of data  0.0
Entity resolution in the web of data    0.0
Automatic acquisition of hyponyms from large text corpora   0.0
Probabilistic Alignment of Relations, Instances, and Schema 0.0
Verbs semantics and lexical selection   0.0
Schema extraction for tabular data on the web   0.0
Dbpedia: A nucleus for a web of open data   0.0
Webtables: exploring the power of tables on the web 0.0
Toward an architecture for never-ending language learning   0.0
Flumejava: easy, efficient data-parallel pipelines  0.0
Probabilistic query expansion using query logs  0.0
Principles of Data Integration  0.0
Open information extraction: The second generation  0.0
Identifying relations for open information extraction   0.0
WordNet: An Electronic Lexical Database 0.0
Incorporating non-local information into information extraction systems by gibbs sampling   0.0
Simple coreference resolution with rich syntactic and semantic features 0.0
Entity search: building bridges between two worlds  0.0
```

图 5

六、基于概率模型的语言模型计算查询 q 与所有文档的相关度

1. 计算并建立所有文档向量和查询向量的语言模型。

对于文档向量、查询向量（以及第二问中文档集）的先验概率的计算，使用的是第 11 章 11.3.3 节中的第 3 种计算方法：

$$p_{t_c} = \frac{1}{3} + \frac{2}{3}\frac{\mathrm{df}_{t_c}}{N_c},$$

$$p_{t_{d/q}} = \frac{1}{3} + \frac{2}{3}\frac{\mathrm{tf}_{t,d/q}}{L_{d/q}}$$

其中 *c*, *d*, *q* 分别代表某个文档集、某个文档和某个查询，*N* 为文档集中的文档数量，*L* 为文档中 token 的数量（去掉停用词之后）。

通过以上方法得到所有文档的语言模型输出到“docs_lan_models.txt”，查询的语言模型为“query_lan_model.txt”，文档集的语言模型为“col_lan_model.txt”，每个文件中每一行都代表 dictionary 里每个 term 的概率（按照 term 编号的顺序）。

2. 计算查询 q 与所有文档的相关度，按相关度从高到低排序后输出。

根据以下计算公式，取λ=1/2，对每个文档计算概率（得分），排序后输出到“lm_rank_and_scores.txt”，格式同图 5。计算过程输出为“lm_calculation.txt”（如图 6），其格式为：每个文档对于每个 query 中的 term 的计算占一行，这一行的格式为：$P(t|M_c)$ $P(t|M_d)$ $1/2(P(t|M_c)+P(t|M_d))$。

$$P(d|q) \propto P(d)\prod_{t\in q}((1-\lambda)P(t|M_c)+\lambda P(t|M_d))$$

```
Online data fusion  0.004811667001606474
Freebase: a collaboratively created graph database for structuring human knowledge  0.004811667001606473
An effective semantic search technique using ontology  0.004774531754626049
Cross-lingual entity matching and infobox alignment in wikipedia  0.0047376831082287145
Learning to link with wikipedia 0.0047376831082287145
Topic indexing with wikipedia  0.0047376831082287145
Result of ontology alignment with RiMOM at OAEI 2006  0.0047376831082287145
Logmap 2.0: towards logic-based, scalable and interactive ontology matching 0.0047376831082287145
Matching large ontologies: A divide-and-conquer approach  0.0047376831082287145
Heterogeneous web data search using relevance-based on the fly data integration 0.0047376831082287145
Federated entity search using on-the-fly consolidation  0.0047376831082287145
Biperpedia: An ontology for search applications 0.0047376831082287145
Computing semantic relatedness using wikipedia-based explicit semantic analysis 0.0047376831082287145
Inconsistencies, negations and changes in ontologies  0.0047376831082287145
Coreference aware web object retrieval  0.0047376831082287145
Repeatable and reliable semantic search evaluation  0.0047376831082287145
Entity recommendations in web search  0.0047376831082287145
Ad-hoc object retrieval in the web of data  0.0047376831082287145
Entity resolution in the web of data  0.0047376831082287145
Automatic acquisition of hyponyms from large text corpora  0.0047376831082287145
Probabilistic Alignment of Relations, Instances, and Schema 0.0047376831082287145
Verbs semantics and lexical selection  0.0047376831082287145
Schema extraction for tabular data on the web  0.0047376831082287145
Dbpedia: A nucleus for a web of open data  0.0047376831082287145
Webtables: exploring the power of tables on the web 0.0047376831082287145
Toward an architecture for never-ending language learning  0.0047376831082287145
Flumejava: easy, efficient data-parallel pipelines  0.0047376831082287145
Probabilistic query expansion using query logs  0.0047376831082287145
Principles of Data Integration  0.0047376831082287145
Open information extraction: The second generation  0.0047376831082287145
Identifying relations for open information extraction  0.0047376831082287145
WordNet: An Electronic Lexical Database 0.0047376831082287145
Incorporating non-local information into information extraction systems by gibbs sampling  0.0047376831082287145
Simple coreference resolution with rich syntactic and semantic features 0.0047376831082287145
Entity search: building bridges between two worlds  0.0047376831082287145
```

图 6

七、对比和分析两种相关度计算的结果，谈谈个人对信息检索相关度计算方面的看法

对比两种相关度计算方法，可以发现计算的结果排名完全一致。这并不是说明它们是一样的效果，而是因为与查询的 terms 有交集的文档仅有 3 篇，故其他的文档完全无关，因此只有 3 篇文档真正被排序了！所以无法比较效果。这个现象的主要原因是查询太短，标题太过简洁。故本人额外使用文档库中的文档《Computing semantic relatedness using wikipedia-based explicit

semantic analysis》构建文档库，来对两种方法进行比较。各种输出文件同上，但放在了“additional_doc”文件夹中。

向量模型的结果如图 7，语言模型的结果如图 8：

```
WikiRelate! Computing semantic relatedness using Wikipedia  9.367092647832465
Indexing by latent semantic analysis    2.861731041693083
Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge    2.7021749743996404
Evaluating wordnet-based measures of lexical semantic relatedness   2.5741742771028124
Extended gloss overlaps as a measure of semantic relatedness    2.3365936110771424
Exploring unexplored contexts for semantic extraction from syntactic analysis   2.163265330135346
Centroid-based document classification: Analysis and experimental results   1.9138557909241556
Corpus-based and knowledge-based measures of text semantic similarity   1.2400442631129542
Feature generation for text categorization using world knowledge    1.1508548153823144
Feature Generation for Textual Information Retrieval Using World Knowledge  1.0765261052698993
Semantic similarity based on corpus statistics and lexical taxonomy 1.071227433146123
Contextual correlates of semantic similarity    0.8735192502636835
Building Large Knowledge Based Systems  0.4861940227697959
Knowledge-Based Systems in Artificial Intelligence  0.4861940227697959
Similarity-based models of word cooccurrence probabilities  0.4438323893253293
An information-based measure and its application to problems of ambiguity in natural language   0.3843701241780805
A web-based kernel function for measuring the similarity of short text snippets 0.3623876283891147
WordNet: An Electronic Lexical Database 0.0
Placing search in context: The concept revisited    0.0
Exploring the similarity space  0.0
Roget's thesaurus as a lexical resource for natural language processing 0.0
Internet encyclopaedias go head to head 0.0
An empirical evaluation of models of text document similarity   0.0
Measures of distributional similarity   0.0
An information-theoretic definition of word similarity  0.0
Roget's Thesaurus of English Words and Phrases  0.0
Contextual correlates of synonymy   0.0
An Introduction to Modern Information Retrieval 0.0
Machine learning in automated text categorization   0.0
Modern Information Retrieval    0.0
```

图 7

```
WikiRelate! Computing semantic relatedness using Wikipedia  0.001191686651004004
Evaluating wordnet-based measures of lexical semantic relatedness   0.0011672697445211852
Corpus-based and knowledge-based measures of text semantic similarity   0.0011655301815988392
Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge    0.0011595221583509565
Indexing by latent semantic analysis    0.0011580858755635046
Extended gloss overlaps as a measure of semantic relatedness    0.0011580858755635046
Exploring unexplored contexts for semantic extraction from syntactic analysis   0.0011580858755635046
Centroid-based document classification: Analysis and experimental results   0.0011578461459378605
Semantic similarity based on corpus statistics and lexical taxonomy 0.0011564315853334307
Feature generation for text categorization using world knowledge    0.0011484227120361738
Feature Generation for Textual Information Retrieval Using World Knowledge  0.0011484227120361738
Contextual correlates of semantic similarity    0.0011473329890680222
Building Large Knowledge Based Systems  0.0011470954853441942
An information-based measure and its application to problems of ambiguity in natural language   0.0011470954853441942
Similarity-based models of word cooccurrence probabilities  0.0011470954853441942
A web-based kernel function for measuring the similarity of short text snippets 0.0011470954853441942
Knowledge-Based Systems in Artificial Intelligence  0.0011470954853441942
WordNet: An Electronic Lexical Database 0.0011380703438387324
Placing search in context: The concept revisited    0.0011380703438387324
Exploring the similarity space  0.0011380703438387324
Roget's thesaurus as a lexical resource for natural language processing 0.0011380703438387324
Internet encyclopaedias go head to head 0.0011380703438387324
An empirical evaluation of models of text document similarity   0.0011380703438387324
Measures of distributional similarity   0.0011380703438387324
An information-theoretic definition of word similarity  0.0011380703438387324
Roget's Thesaurus of English Words and Phrases  0.0011380703438387324
Contextual correlates of synonymy   0.0011380703438387324
An Introduction to Modern Information Retrieval 0.0011380703438387324
Machine learning in automated text categorization   0.0011380703438387324
Modern Information Retrieval    0.0011380703438387324
```

图 8

前 17 篇文档都是不完全无关的（有与 query 的 terms 重叠的词）。这次的结果并不完全相同。这种差异可能是与两种模型对于文档长度在排序中的使用以及文档集中 term 频率的不同使用方法有关：向量模型使用的是 idf，而语言模型中使用的则是 cf。而由于语言模型中使用的是 unigram 模型，故语序并没有起作用，因此，本实验中的语言模型与向量模型所使用的其他信息比较相似，两种模型的结果也差不多：query 中出现的词在 document 中出现得越多，这个 document 就越靠前，差异仅体现几个 document 之间具有相同的 query 词频的时候。

而实际应用中不能忽视语序，故对于语言模型，最好使用 bigram 及以上的模型，否则效果与向量模型差不多。